Media Guardian Prototype Documentation

---

1. Data Scraping:

- We start by scraping data from various news websites. We identify the target websites, here we choose some popular news websites from which we want to scrape data.
- We use web scraping tools or libraries like Beautiful Soup, Scrapy, or requests to extract data from the websites. We store the scraped data in a structured format, like JSON or a database, for further processing.

2. Search Functionality:

- Our website has a search feature for users to input a policy or topic of interest.
- We have implemented a search feature in the front-end (react app), and it can communicate with the backend using API requests.

3. Backend Server:

- We set up a backend server, which is built using Flask (Python). The server will take policy requests from the front end and give the scraped data a response.

-For youtube and twitter it will use suitable apis provided by them to scrape data and those also send a json response where every state of India is a key and and the scraped data is the value.

-This scraped data will go through the data cleaning process and then will be given to a trained machine learning model and feedback from various states and the public will be given as a response.

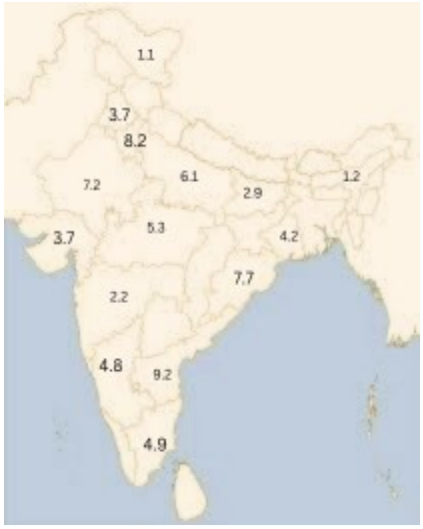
- Backend apis will be hosted in AWS.

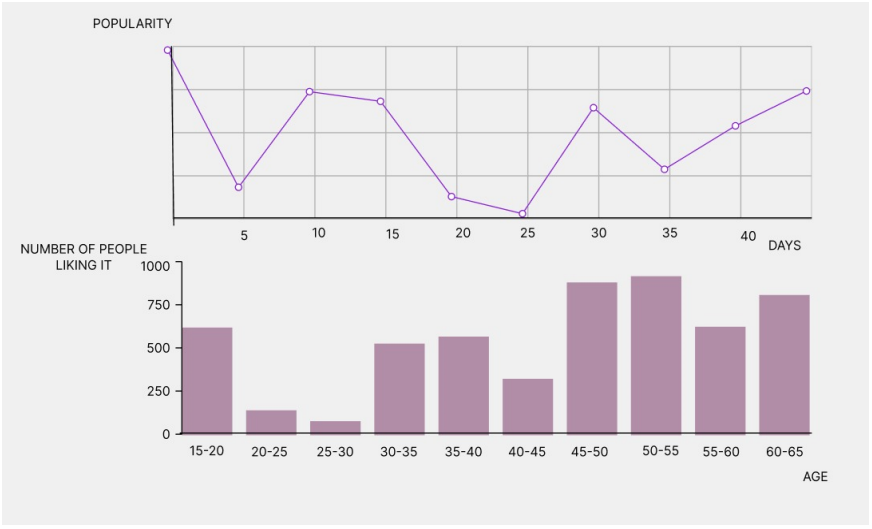
4. Front-End Interface:

- In frontend interface - we organize the data came from backend into a tabular format to represent scraped data where the source of the scraped data will be shown.
- We display the YouTube transcript data,twitter data alongside the web-scraped data in a readable format on our website.

-If we click any scraped data link our dashboard takes this data and transforms it into actionable visualizations. The first, a dynamic sentiment graph, unveils the intricate relationship between public sentiment and the passage of time. This temporal analysis enables users to pinpoint shifts in sentiment trends, recognize overarching patterns, and respond promptly to critical changes in public perception.

The second visualization, presented on an interactive Indian map, delves into the finer details of sentiment variations. This geographical perspective offers a state-wise breakdown of public sentiment, helping policymakers navigate the complexities of regional dynamics. By dissecting sentiment trends across the map, our dashboard equips decision-makers with a powerful tool for crafting targeted policies that resonate with diverse regional populations.





5. Machine Learning Pipeline:

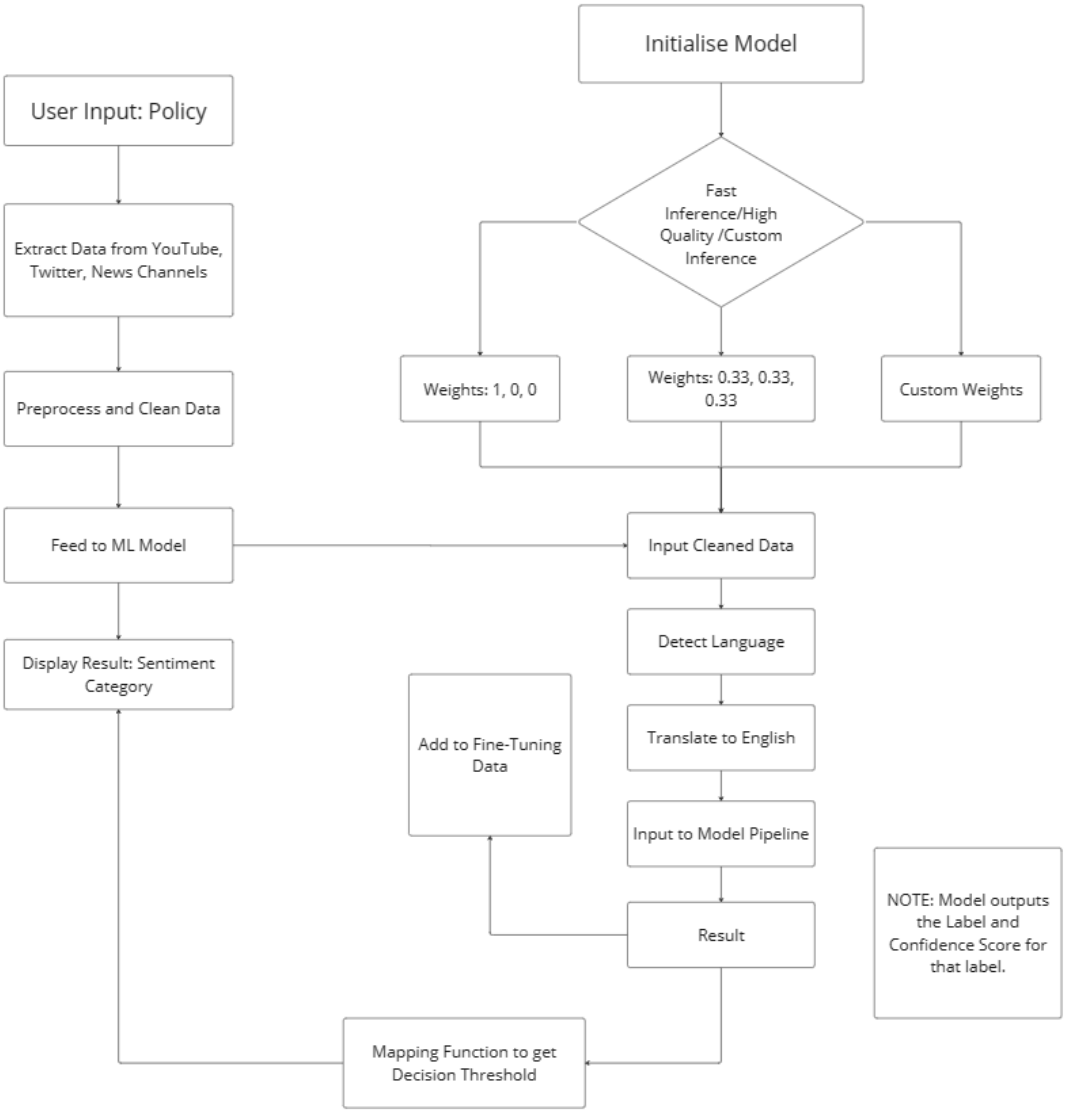
The Pipeline detects language and translates the text into English and splits large texts into smaller chunks for effective processing by the model. We have fine-tuned Roberta Base Uncased for Sentiment Analysis. This model returns predicted class (0 for negative, 1 for neutral and 2 for positive) and confidence of that prediction.

We use a mathematical mapping to map the class and confidence to a Sentiment Score on a continuous scale of -1 to 1, where -1, 0, and 1 correspond to negative, neutral and positive sentiment respectively.

An aggregate of these scores is used to get the overall sentiment for each source of all states/regions, which is then displayed on the website.

For the above processes, we only use local modules and models, no third party API is used.

6. Data Flow:


Initialise Model
User Input: Policy
Fast Inference/High Quality /Custom Inference
Extract Data from YouTube, Twitter, News Channels
Weights: 1, 0, 0
Weights: 0.33, 0.33, 0.33
Custom Weights
Preprocess and Clean Data
Feed to ML Model
Input Cleaned Data
Detect Language
Display Result: Sentiment Category
Add to Fine-Tuning Data
Translate to English
Input to Model Pipeline
NOTE: Model outputs the Label and Confidence Score for that label.
Result
Mapping Function to get Decision Threshold